NEXO: Adaptive Visualization for Comparative Exploration of Knowledge Graphs with Natural Language Interaction

Reza Shahriari, Justin Whorton, Jonathan Bona, Kevin Sexton, Mathias Brochhausen, Jaime Ruiz, and Eric D. Ragan

Abstract—Knowledge graphs (KGs) capture rich, multi-relational data, but comparing entities, relationships, and local structures remains challenging with traditional graph visualizations. We present NEXO, a task-driven system for comparative KG exploration that combines natural language interaction with adaptive, coordinated visualizations. The system translates free-form queries into structured comparison tasks and supports value comparison, path comparison, and neighborhood exploration across one-to-one, one-to-many, and many-to-many comparison structures. Rather than exposing users to the full graph, NEXO generates task-specific views that combine structural context with concise summaries to support efficient comparison. We evaluate the system through a user study with eye tracking to assess task performance and attention allocation across coordinated views. Participants completed tasks with high accuracy and minimal query reformulation. Gaze patterns reveal task-dependent attention strategies, with users relying more on graph views for structural exploration and summary views for pattern identification and comparison. These findings demonstrate the effectiveness of combining natural language interaction with task-adaptive visualizations for comparative KG exploration.

Index Terms—Knowledge Graph, Graph Visualization, Natural Language Interaction, Human-in-the-Loop

1 Introduction

Network data represents interconnected entities within various systems, highlighting their relationships and dependencies. A knowledge graph (KG) is a specific type of graph that organizes information by linking entities and their attributes within a structured, semantic framework. KGs are widely used across domains due to their ability to encode semantically rich information [16, 25, 30, 43, 44]. For example, node–link diagrams are commonly used for KG visualization, but they suffer from scalability limitations that hinder effective exploration of large-scale graphs [22]. Although numerous graph visualization techniques have been proposed to mitigate these challenges, most approaches are designed for general network data and are not specifically tailored to the semantic structure and relational complexity of KGs [20, 51]. Importantly, KGs support a wide range of analysis tasks by combining multidimensional data, such as entities, relationships, and attributes, within a single representation. However, this richness also introduces complexity, making it difficult for any single visualization technique to effectively support all task types.

While a few systems have attempted to adapt these techniques for KGs [7, 15, 27], a substantial gap remains between traditional network visualization approaches and the unique requirements of KG visualization. In particular, limited attention has been given to natural language interaction (NLI) as a mechanism for exploring KGs, despite evidence that language-based interfaces can improve usability and accessibility in data visualization contexts [23, 28, 31, 35, 37, 39, 41]. Allowing users to interact through natural language, while accounting for the inherent ambiguity of such inputs [12], has been shown to support effective and flexible data exploration across various scenarios, including network datasets [35, 41]. However, most existing work on NLI for visualization has focused primarily on tabular data and statistical charts, leaving interaction with more complex, heterogeneous, and relational structures such as KGs largely underexplored.

- *Reza Shahriari, Jaime Ruiz, and Eric D. Ragan are with the University of Florida. E-mail: {rshahriari | jaime.ruiz | eragan}@ufl.edu.*
- *Justin Whorton, Jonathan Bona, and Mathias Brochhausen are with the University of Arkansas for Medical Sciences. E-mail: {jmwhorton | jpbona | mbrochhausen}@uams.edu.*
- *Kevin Sexton is with Vanderbilt University Medical Center. E-mail: kevin.sexton@vumc.org.*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

Our research also aims to address common challenges associated with simultaneously exploring and comparing different portions of a larger graph. Comparison is a fundamental analytic operation in KG exploration, as users often need to contrast entities, relationships, or local graph structures to identify similarities, differences, and patterns [13]. However, performing such comparisons in KGs is challenging, as relevant entities and relationships are often distributed across multiple parts of the graph and difficult to inspect simultaneously. Additionally, despite advances in interaction techniques, existing research has rarely focused on applying these methods to comparison tasks in KG. To bridge this gap, we propose applying visualization and interaction techniques specifically to support comparison tasks in KG exploration with natural-language interaction. Our work focuses on three primary types of comparisons: one-to-one, one-to-many, and many-to-many. These comparison types align with three common graph-related task categories inspired by prior work [21]: value comparison, path comparison, and neighborhood exploration. By explicitly supporting different types of comparison tasks, we aim to cover common query scenarios to enhance users' ability to parse and grasp information needed to answer their task at hand efficiently without being overwhelmed by the scale of too much data.

Thus, our primary contribution is NEXO, a comparison-driven visual analytics system for KG exploration (Figure 1). Rather than using natural language solely for graph query generation, NEXO combines comparison-oriented task formulation, human-in-the-loop query interpretation, and adaptive coordinated visualizations to support value comparison, path comparison, and neighborhood exploration. NEXO is designed around three key goals: (DG1) enabling natural language interaction for expressing comparison intent, (DG2) supporting task-driven comparison through adaptive graph visualizations, and (DG3) enabling multi-level comparison of graph context and statistical summaries. We suggest that comparison is an important yet often under-supported operation in KG exploration, and that addressing it may benefit from a task-driven integration of natural language interaction and adaptive multi-view visualizations. We demonstrate the efficiency of our approach through the implementation of a system prototype, with particular attention to scalability, ensuring that large graphs with complex node and edge structures can be handled interactively.

2 Background

2.1 Knowledge Graph Visualization

KGs illustrate the connections between entities, enabling an understanding of relationships. Various visualization formats support in-

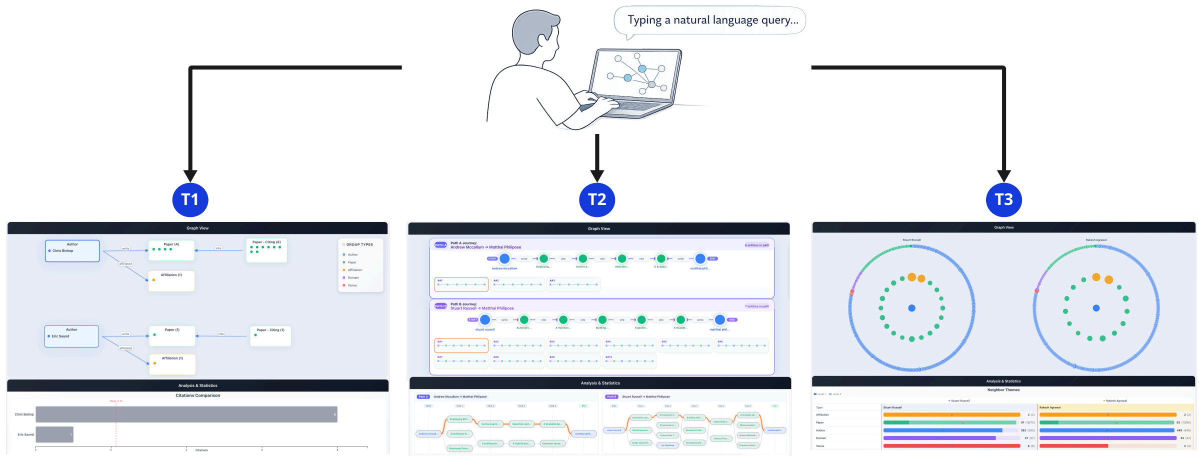

Figure 1: System interface supporting comparison-driven KG exploration. Users issue natural language queries (top), which are interpreted into three task types: T1 value comparison, T2 path comparison, and T3 neighborhood exploration. Each task is presented through adaptive coordinated views, including a graph view for structural reasoning and an analysis and statistics panel for summarizing and comparing results.

terpretation, with node-link diagrams being the most common, using points (e.g., circles or boxes) for entities and lines for connections [26]. Another common representation is adjacency matrix [26], where nodes are arranged along the vertical and horizontal edges of a square, and connections are indicated by flagging the cell at the intersection of two nodes. Moreover, multiple tools [10, 17] combined these representations to allow a hybrid representation of network data that can take advantage of multiple representations. For instance, NodeTrix [17] combines node-link diagrams and adjacency matrices into a single, cohesive representation. The global structure of the network is displayed as a node-link diagram, while communities, which are often dense and difficult to interpret using only node-link representations, are shown as matrices.

Although the most common approach is the use of node-link diagrams, prior research has identified several limitations to this method, particularly in large or dense graphs. Li et al. [22] show that node-link diagrams often suffer from visual clutter and scalability issues, which can hinder users' ability to perform effective exploration and sensemaking. To address these limitations, researchers have explored alternative visualization techniques tailored to specific analytic tasks. For example, systems like TreePlus [21] and Pathfinder [33] redesign graph layouts specifically for path exploration tasks. TreePlus [21] converts the graph into a tree layout rooted at a selected node, making relationship paths easier to follow by reducing visual clutter. Pathfinder [33], on the other hand, dynamically reorders and bundles edges along selected paths to maintain spatial continuity, helping users visually track sequences of connected nodes in dense graphs.

Despite advances in KG visualization, comparison remains relatively underexplored as a primary design objective. Prior visualization research has established comparison as a fundamental analytical activity. For example, Tominski et al. [46] proposed interaction techniques inspired by natural comparison behaviors to support the selection, arrangement, and comparison of visual information across views. Similarly, Alper et al. [2] investigated weighted graph comparison and demonstrated the importance of dedicated visual representations for identifying structural differences across graph datasets. Also, Gleicher et al. [13] further emphasize that comparison is fundamental to visual analysis. While these works provide important foundations for comparison-oriented visualization, they do not explicitly support structured comparison across entities, paths, and neighborhoods in KGs. Furthermore, the integration of natural language interaction with comparison tasks in graph settings remains limited, despite its growing adoption in tabular and statistical visualization interfaces [39].

2.2 Natural Language Interaction for Visualization

Natural language interfaces for data visualization have emerged as a promising approach to make data exploration more accessible by allowing users to express analytic intentions using natural language instead of complex interfaces or programming [39]. Many systems have explored natural language interaction for visualization by addressing different forms of user intent and ambiguity [12, 28, 35, 41, 50]. For example, DataTone [12] introduced a mixed-initiative approach that translates user queries into visualizations while surfacing ambiguity through interactive widgets, such as dropdowns and sliders, that help users refine their intent without needing to rephrase their input. Building on this idea, Eviza [35] supported more dynamic, context-aware interaction by allowing follow-up questions and integrating natural language with on-chart widgets, enabling users to clarify vague or underspecified queries over multiple conversational turns. Also, Orko [41] explored a multimodal interaction paradigm specifically for network data exploration, where users issue speech commands alongside touch gestures on a tablet interface to navigate, filter, and manipulate node-link visualizations, supporting tasks such as path tracing and neighborhood inspection. More recently, natural language interfaces have been extended to graph databases. For example, Ozsoy et al. [32] introduced Text2Cypher, which translates natural language questions into executable Cypher queries, reducing the need for users to learn graph query languages. While such approaches improve access to graph data, their primary focus is query generation and information retrieval rather than supporting visual analysis tasks.

While these systems demonstrate the effectiveness of natural language interaction for visualization and graph querying, they primarily focus on data retrieval, navigation, or general exploration tasks. Support for structured comparison remains limited, particularly in KG settings where users must compare entities, relationships, and local graph structures. This gap motivates the need for systems that combine natural language interaction with comparison-oriented visual analysis.

3 Comparative Knowledge Graph Exploration

While Section 2 reviewed prior work on KG visualization and NLI, this section introduces the conceptual framework that motivates the design of NEXO. Our framework builds upon foundational work in graph analysis and visual comparison, which remains relevant because it defines general analytical operations, such as comparison, path reasoning, and neighborhood inspection, that are independent of a particular visualization system or application domain. We leverage these foundations to formulate comparison tasks specifically for KG exploration.

Comparison is a central operation in visual analysis, enabling users to identify similarities, differences, and patterns that support higher-

Table 1: Comparison task types and example scenarios (one representative task per cell), spanning value, path, and neighborhood comparisons across different comparison structures.

	Comparison structure		
Task type	**1–1 (entity–entity)**	**1–N (entity–set)**	**N–N (set–set)**
T1 **Value comparison**	Does Chris Bishop have more citations than Eric Saund?	Does Wanna Iba have more citations than *all* authors affiliated with the University of Turin?	Do authors affiliated with University of Milan have more total citations than authors affiliated with University of Miami?
T2 **Path comparison**	Is the shortest path from Stuart Russell to Andrew McCallum longer than the path from Stuart Russell to Matthai Philipose?	Does Ali Rahimi have more shortest paths to authors affiliated with Arizona State University than to authors affiliated with Intel?	Which affiliation group (Arizona State University vs. Intel) is, on average, closer to Ali Rahimi in the coauthorship network?
T3 **Neighborhood exploration**	Is Stuart Russell directly connected to more papers than Rakesh Agrawal?	Within depth 2, do authors affiliated with the University of Turin show denser collaboration than Ali Rahimi?	Within depth 2, which group (University of Georgia vs. Brandeis University) publishes in a broader set of research domains?

level reasoning and decision-making [3, 13]. In the context of KGs, comparison is particularly critical given the inherently relational and multi-entity nature of the data: users often need to simultaneously compare entities, relationships, and local structures to understand influence, connectivity, or structural roles. However, this type of reasoning is cognitively demanding, as it requires coordinating information across multiple subgraphs, tracing connections both within and between entities, and integrating structural and attribute-based evidence.

While prior research in graph visualization and network analysis has explored node-link layouts, clustering, and interactive techniques for navigation and local inspection [1, 4, 45], these approaches primarily support exploration and connectivity analysis rather than explicitly enabling structured comparison across entities and relationships. As a result, users are often left to manually assemble comparisons across views, leading to overload and reduced efficiency, especially for complex tasks involving multiple entities or subgraphs.

To address this gap, we formalize comparison in KGs through a task-oriented framework grounded in fundamental forms of network or graph data analysis. Through conceptual analysis of how users contrast entities and relationships, we identify three complementary modes of reasoning: attribute-based reasoning over nodes, relational reasoning over paths, and structural reasoning over local neighborhoods. These correspond to value comparison, path comparison, and neighborhood exploration. This task formulation constitutes a core conceptual contribution of our work, providing a structured basis for both the interpretation of comparison intent from natural language and the design of task-adaptive visual representations. By explicitly connecting comparison tasks to both natural language interpretation and visualization requirements, the framework guides how comparison results are extracted, represented, and explored within NEXO.

Our research follows an organization of KG comparison tasks building on established visualization task taxonomies and extends them to the relational and multi-entity nature of KGs. We draw upon established task taxonomies in graph and information visualization [3,21,40], which characterize fundamental analytical operations such as comparison, connectivity reasoning, and structural analysis that remain central to graph exploration despite advances in visualization techniques and interaction paradigms. However, KGs introduce richer semantics and multi-relational edges, requiring comparison across nodes, heterogeneous relationships, and local structures. Through iterative conceptual refinement, we identify three core comparison tasks (Table 1) that capture these requirements and are broadly applicable to KG-based analysis:

- **Value Comparison**: comparing numerical or categorical attributes across nodes (e.g., citation count, domain, affiliation).
- **Path Comparison**: evaluating relationship paths between entities (e.g., shortest path, number of intermediaries).
- **Neighborhood Exploration**: examining differences in local graph structure or neighborhood patterns.

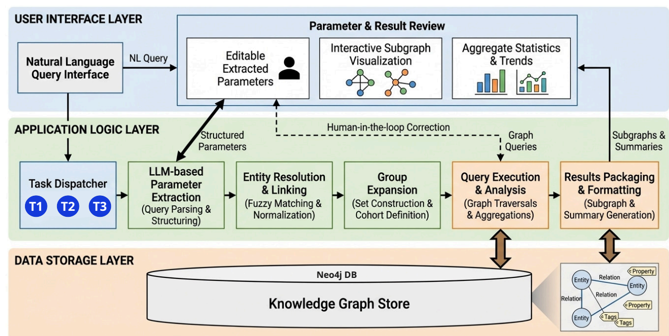

Figure 2: System architecture: The user interface layer supports NL query input, parameter review, and coordinated visual outputs. The logic layer parses queries using an LLM-based pipeline, performs entity resolution, constructs comparison groups, and executes graph queries.

Each of these tasks can involve different comparison structures:

- **1–1 Comparison**: comparing two specific entities.
- **1–N Comparison**: comparing one entity to a group.
- **N–N Comparison**: comparing across two groups of entities.

This formulation remains consistent with prior taxonomies while extending them to support comparison in multi-relational settings. We further validate this decomposition by mapping representative queries to these categories, confirming that they capture the comparison scenarios supported by our system. We use this task framework to guide the system design, where each task type informs how natural language queries are interpreted and how results are represented through coordinated visualizations.

4 System

To address the comparison tasks identified in Section 3, we developed NEXO, a visual analytics system for comparison-driven exploration of KGs through natural language interaction (Figure 2). The system combines three key ideas: a comparison-oriented workflow for KG exploration, transparent human-in-the-loop query interpretation, and task-adaptive coordinated visualizations that support different forms of comparison.

4.1 Design Goals

We derive our design goals directly from comparison task framework introduced in Section 3. Our goals support three key stages of comparison: expressing comparison intent (aligned with task specification), visualizing comparison structures (aligned with task execution), and interpreting results across representations (aligned with task reasoning).

DG1. Enable Natural Language Interaction for Flexible Comparison

Figure 3: System interface for T1: Value Comparison. The interface follows a consistent multi-panel design used across all task types (T1–T3). A shows the chat interface where users issue natural language queries. B presents extracted parameters and supports ambiguity handling through human-in-the-loop refinement, allowing users to verify and edit the system's interpretation. C provides textual summaries and intermediate results to support quick understanding. D displays the graph view for structural reasoning over entities and relationships. E presents the analysis and statistics panel, summarizing comparison results through concise visual encodings.

Figure 4: Analysis & Statistics Panel for T1 Value Comparison. The panel presents comparative summaries for different comparison structures, including 1–1, 1–N, and N–N scenarios. Bar charts show aggregated values (e.g., number of papers), while reference lines indicate summary statistics such as mean and median. This view enables users to quickly compare values across entities or groups without inspecting the full graph, supporting efficient interpretation of differences across comparison types.

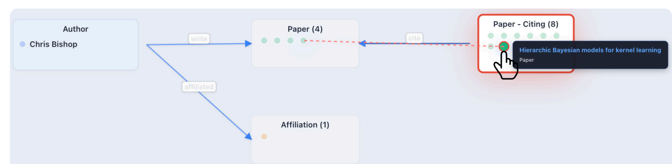

Figure 5: Interaction for T1 : Value Comparison. Hovering over a node highlights its related entities and reveals underlying relationships using dotted links, indicating which papers cite or are cited. This interaction allows users to trace the origin of aggregated values and verify how comparison results are derived.

Recent research highlights the potential of Visualization-oriented Natural Language Interfaces (V-NLIs) to lower the entry barrier to data analysis by allowing users to express analytical intent through natural language rather than complex tool-specific operations [12, 39]. However, much of the existing work on V-NLIs has focused on tabular data and statistical visualizations [9, 28, 39], leaving interaction with more complex data structures such as KGs relatively underexplored. KGs represent richly interconnected data that can be difficult to navigate and query, particularly for users unfamiliar with graph query languages. Our goal is to support exploration and comparison of graph structures, relationships, and attributes through a combination of natural language interaction and direct manipulation, without requiring specialized technical knowledge. Because natural language input can be ambiguous and imprecise, the system should support transparent interpretation of extracted intent and allow users to review, refine, and resolve ambiguities before analysis is performed [12]. Rather than treating natural language interaction as a black-box query mechanism, our goal is to expose and make editable the intermediate interpretation used to generate visual analysis results.

DG2. Task-driven Comparison through Adaptive Graph Views

Prior work shows that task-driven design enhances users' analytic capabilities by aligning visualization systems with the activities users perform during analysis, including core tasks such as comparison [3, 42, 49]. In the case of graph-based data, where users may need to compare complex topological structures, node attributes, or relational patterns, enabling flexible, adaptive graph views can help parse information for supporting tasks such as side-by-side comparisons, highlighting differences, or visualizing alternative subgraph selections. Our goal is to build on this literature to create a system that dynamically adapts its visual outputs based on the comparison task at hand, whether explicitly requested through natural language or inferred from interaction context.

DG3. Supporting Multi-Level Comparison of Graph Context and Statistics

Effective data exploration often requires users to shift between overview and detail views [8, 40]. Multi-level interfaces improve analysis by allowing users to maintain context while inspecting specific nodes, relationships, and statistics [34, 38, 40]. This is particularly important for comparative analysis, where users must understand both overall patterns and the details underlying those patterns [8, 18]. In graph exploration, combining structural overviews with statistical summaries enables efficient identification of differences and trends while preserving access to quantitative information. Furthermore, brushing and linking across coordinated views helps users trace entities and maintain continuity between levels of abstraction [34, 48].

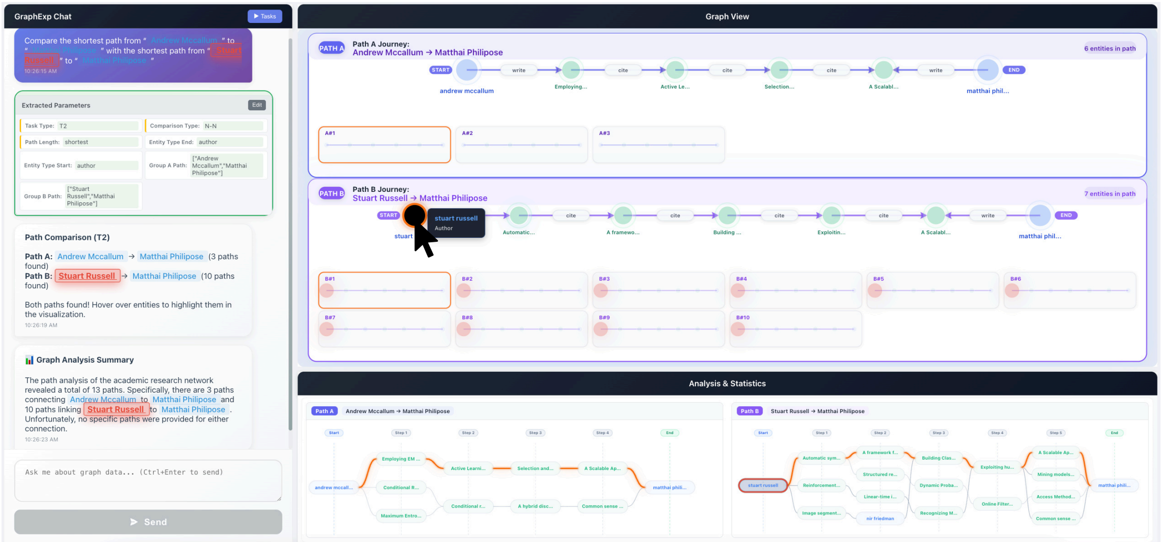

Figure 6: Brush-and-link interaction for T2: path comparison. When users hover over a node in the graph view, the system highlights the corresponding entity across all coordinated views, including text and nodes in other panels. This interaction enables users to trace connections, inspect intermediate entities, and compare multiple paths without losing context.

4.2 Usage Scenario with design overview

Our system supports comparative exploration of KGs by allowing users to express analysis questions in natural language and automatically translating them into structured graph queries and visual comparisons. Instead of requiring users to manually navigate complex node-link diagrams or write formal graph queries, the system interprets natural language input to identify the intended comparison task and extract the relevant parameters. Based on this interpretation, the interface adaptively presents results through coordinated panels that support different task types and explicitly handles multiple comparison types.

To illustrate the system in use, consider a researcher exploring a scholarly KG dataset such as the KG20C dataset [47]. This dataset contains papers from 20 top-tier computer science conferences and represents scholarly relationships as a multi-relational graph with five entity types (*paper*, *author*, *affiliation*, *venue*, and *domain*) and five relation types (*author_in_affiliation*, *author_write_paper*, *paper_in_domain*, *paper_cite_paper*, and *paper_in_venue*). Suppose the researcher is investigating differences in research impact and publication patterns between two authors, aiming to understand not only who is more cited but also how their work differs across domains, venues, and citation behavior, as a starting point for exploring broader structural and relational differences in the graph. To begin, they ask: *"Compare citations of Chris Bishop and Eric Saund."* The system interprets this as a value comparison task (T1) and a 1-1 comparison type, and extracts the relevant parameters. The system presents these parameters to the user in the chat interface so they can verify or edit the interpretation if needed (Figure 3). Also, the visualization panel presents a concise comparison of citation counts (Figure 4), while the graph view highlights the corresponding authors and their connected publications, allowing the user to inspect where the citation values originate (Figure 5).

Building on this initial comparison, the researcher next wants to understand whether the difference in citation impact may be related to how each author is connected to influential researchers in the broader community. After observing that one author appears to have stronger visibility in the dataset, they ask whether this difference may reflect closer connections to a prominent researcher. To do this, they type *"Compare the shortest path from Chris Bishop to Stuart Russell with the shortest path from Eric Saund to Stuart Russell"* to examine how directly each author is connected to Stuart Russell, an influential researcher in the field (Figure 6). The system interprets this as a path comparison task (T2) and retrieves the relevant paths in the KG. The graph view highlights the candidate paths as node-link chains, allowing the researcher to inspect the intermediate collaborators, papers, or venues connecting each author to Stuart Russell. At the same time, the visualization panel summarizes the paths in a structured layout, making it easier to compare the two connections and assess whether one author is more directly linked.

Finally, after comparing how each author connects to Stuart Russell, the researcher seeks to understand whether these connections are part of broader differences in their surrounding research communities. In particular, they want to examine whether the authors are embedded in different neighborhoods of collaborators, venues, or research domains that may explain the structural differences observed in the paths. To do so, they ask: *"Show me nodes around Stuart Russell and Rakesh Agrawal for two levels around it."* The system interprets this as a neighborhood exploration task (T3) and extracts the relevant parameters, including the target entities and the requested depth. The graph panel visualizes the local neighborhoods using radial tree layouts centered on each author, revealing surrounding collaborators, papers, venues, and domains. In parallel, the visualization panel summarizes these neighborhoods in a comparative table (See T3 in Figure 2), showing the number and types of neighboring entities across the two levels of expansion. This allows the researcher to quickly identify differences in collaboration patterns and research areas without manually inspecting each node-link structure. To further investigate specific connections, the researcher can directly interact with the graph by expanding nodes and edges on demand, enabling progressive exploration while maintaining the overall comparative context.

4.3 System Architecture

NEXO follows a web-based client–server architecture designed to support interactive KG exploration through natural language interaction. The system consists of a React-based frontend, a Python (Flask) backend, and a Neo4j graph database (Figure 2). This separation allows dynamic query processing, graph retrieval, and responsive visualization. The frontend presents a triple-panel interface (Figure 3). The left panel provides a conversational interface for natural language input. The right side contains two coordinated views: an interactive node–link graph (top) and a task-specific analysis panel (bottom). Graphs are rendered using D3.js [5], with visual encodings for entity types and relations. The analysis panel generates structured summaries (e.g., value comparisons, extracted paths, or neighborhood statistics) depending on task type. All frontend–backend communication is handled asynchronously via RESTful API calls, allowing iterative refinement.

On the back end, incoming queries are processed through a structured pipeline. Rather than directly generating graph queries from free-form input, the pipeline first translates each natural language request into an intermediate structured specification. This specification explicitly encodes the comparison task type (T1/T2/T3), comparison structure (e.g., 1–1, 1–N, or N–N), referenced entities or groups, and task-specific parameters such as aggregation targets, path constraints, or neighborhood depth. The structured specification serves as an intermediate representation between the natural language interface and the backend. This specification also serves as the primary mechanism for context management within NEXO. Rather than relying on implicit conversational memory, the system externalizes the current interpretation state through a structured, user-visible specification that captures the comparison task, entities, comparison structure, and task-specific parameters. This state remains available throughout the analysis process and can be directly inspected and modified during iterative refinement.

Natural language input is analyzed using `gpt-4.1` [29], which is used for both parameter extraction and summary generation. The system message contains a structured prompt specifying task classification rules (T1/T2/T3), few-shot examples, multilingual handling, and an explicit no-hallucination constraint requiring the extraction of only explicitly mentioned entities. To reduce unsupported interpretations, the extraction stage follows several conservative safeguards. The model is instructed not to invent entities, not to infer missing parameters unless they are directly supported by the query, and not to collapse underspecified requests into a fully executable interpretation. When a query is incomplete or ambiguous, the system preserves unresolved fields in the extracted specification and exposes them to the user for verification rather than silently guessing a value.

Following LLM extraction, entity resolution is performed in two stages. First, a fuzzy matching module compares extracted entity names against Neo4j entries. If similarity exceeds a threshold of 0.8 (tuned empirically to reject similar but distinct names while catching the minor typos), the entity is automatically corrected; otherwise, the original term is retained. Task-specific extractors then compute results for value (T1), path (T2), or neighborhood (T3) comparisons. The backend returns structured results, including resolved parameters, computed comparisons, and relevant subgraphs, for rendering in the graph and analysis panels. The pipeline supports iterative refinement through an explicit, editable parameter specification shown in the interface. After each successful query, the extracted task type, comparison structure, and task-specific parameters are exposed to the user for inspection and modification. If users adjust these parameters directly, the system bypasses the LLM and re-executes the selected task deterministically from the updated specification. This design makes the system interpretation visible and correctable without requiring users to restate the full query. We must also note that the mean end-to-end *response time* during the user study was 2.11 seconds, which shows the pipeline worked fast enough to avoid major disruptions to the user's analysis. This architecture supports a closed feedback loop in which users can iteratively refine queries or combine natural language input with direct graph interaction.

4.3.1 Ambiguity Handling

Prior research has highlighted ambiguity as a fundamental challenge in natural language interfaces for visualization, where users issue underspecified and flexible queries during exploratory analysis [12, 41]. Enabling natural language interaction with KGs using our system, therefore, requires systematic resolution of ambiguity across task intent, entity reference, and correct extraction. In practice, ambiguity appears in several distinct forms. *Task ambiguity* arises when a query could plausibly describe a value comparison, a path comparison, or a neighborhood exploration task. *Entity ambiguity* occurs when names are misspelled, abbreviated, or refer to multiple similar KG entities. *Parameter ambiguity* appears when the request omits values needed for execution, such as depth, path length, or aggregation target. Finally, *scope ambiguity* can arise in follow-up queries when users intend to refine only one part of a previous request rather than issue a completely new analysis.

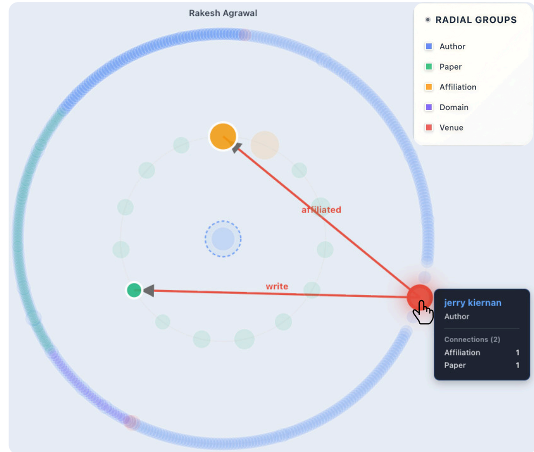

Figure 7: Interaction with radial tree for T3 : Neighborhood Exploration. Hovering over a node reveals its connected edges and highlights related entities, while displaying detailed information such as connection types and counts. This interaction supports inspection of local structure and reduces visual clutter by exposing details on demand.

To address this, our system implements a multi-stage pipeline that resolves ambiguity through constrained semantic extraction, database-grounded normalization, and human-in-the-loop (HITL) feedback [36] for refinement (See Figure 2). Free-form queries are first transformed into structured task specifications (T1: value comparison, T2: path comparison, T3: neighborhood exploration) using an LLM-based extractor equipped with strict prompt-level safeguards. A hard no-hallucination rule ensures that only explicitly mentioned entities are extracted, while flexible phrasing recognition and few-shot task examples map diverse linguistic to the task representations.

After parameter extraction, each named entity is matched against the Neo4j database to ensure it corresponds to an existing node. Matching is performed using a deterministic similarity score that combines (1) character-level sequence alignment, (2) substring containment, and (3) word-level overlap. If the final score is ≥ 0.8, the extracted string is automatically replaced with the canonical database name; otherwise, it is left unchanged to avoid incorrect substitutions.

The final parameter specification, including any corrections, is then rendered directly in the interface as a structured, editable object. Each field (task type, entities, aggregation, depth, path length) can be modified inline or through an editor panel (see Figure 3 B). If a user edits the parameters, the system bypasses the LLM entirely and executes the analysis deterministically from the updated specification. This separation between semantic interpretation and execution allows users to correct extraction errors in one or two interactions without reissuing the natural language query. This grounding step maps extracted mentions to canonical KG node labels before execution, ensuring that comparison operations are performed over verified graph entities rather than raw text strings.

Unlike prior systems that resolve ambiguity primarily through visualization-layer widgets, we found that such widgets are not sufficient for KG queries where ambiguity often arises before visualization (e.g., wrong task type, incorrect entity, or missing structural parameters). In our setting, ambiguity must be resolved at the semantic and database levels before meaningful graph results can be generated. Therefore, we handle ambiguity during extraction and entity normalization, and use structured parameter editing as a direct correction mechanism. The interface exposes the full extracted specification and allows users to edit any field explicitly, rather than adjusting inferred values indirectly through visualization controls. This approach makes errors easier to detect and correct, avoids repeated dialog interactions, and ensures correct execution while maintaining low response latency (M = 2.11s).

4.4 User Interface

NEXO presents a coordinated three-panel interface that integrates natural language interaction, graph visualization, and task-specific summaries (Figure 1). The layout consists of a *chat panel*, central *graph view*, and *analysis and statistics* panel, supporting iterative exploration.

The *chat panel* serves as the primary interaction entry point. Users submit natural language queries and receive formatted responses within a threaded message view. After each successful query, the system injects a structured *parameter extraction panel* summarizing extracted and auto-corrected fields, including task type, comparison type, entities or groups, aggregation settings, path length, and depth. Each parameter is editable inline or through a modal editor supporting form-based controls and validated JSON editing. Editing parameters re-executes the analysis deterministically without re-invoking the LLM. Entity names appearing in system responses are clickable and trigger coordinated highlighting across panels through *brush-and-link* interactions. This design exposes the semantic interpretation of each query, making ambiguity visible and correctable.

The *graph panel* renders the academic KG using D3.js [5] with a force-directed layout, pan, zoom, and interactive tooltips. Node size is logarithmically scaled by degree, while color encodes entity type (e.g., author, paper, affiliation). Interaction behaviors are task-dependent. For value comparison (T1), subjects are displayed in expandable grouped regions; expansions are reversible at the node level (Figure 3D). For path analysis (T2), paths are rendered as parallel node-link chains with distinct colors and selectable alternatives (Figure 6). For structural comparison (T3), entities are visualized as radial trees rooted at the subject, with breadth-first depth encoded through concentric rings and opacity (Figure 7). Newly expanded nodes are animated to indicate change. A reset control restores the overview graph. These task-specific layouts preserve structural clarity while maintaining consistent interaction primitives.

The *analysis and statistics panel* provides task-specific summaries linked to the graph view. For T1, horizontal bar charts support 1–1, 1–N, and N–N comparisons with color-coded roles and reference statistics. For T2, side-by-side path diagrams display selected paths with synchronized highlighting. For T3, a neighbor-themes table summarizes breadth-first expansions by entity type and depth using stacked horizontal bars with opacity encoding depth level. All visual summaries are brush-and-link coordinated: hovering or selecting an entity in any panel highlights corresponding elements across the interface, ensuring consistent state propagation between chat messages, graph nodes, and statistical summaries.

5 Evaluation

Our evaluation is designed to assess the key claims of our approach, as articulated through the design goals in Section 4.1. We report both quantitative and qualitative results, examining task performance, interaction behavior, and gaze patterns.

5.1 Study Design and Procedure

We conducted a controlled lab study to evaluate how participants use our system to perform comparison-driven exploration tasks on a KG using natural language interaction. The study was designed with three **primary objectives**: (1) to assess the overall usability of the system, (2) to analyze how participants distribute attention across the coordinated panels, examining how panel usage and preference vary depending on the task type and comparison structure, particularly when multiple levels of detail are available, and (3) to gather qualitative feedback on the system's features, interaction design, and effectiveness for supporting comparison tasks.

We initially aimed to conduct a comparative evaluation against existing systems. However, we were unable to identify publicly available KG exploration tools that support both natural language interaction and task-driven comparison in a manner comparable to our system. Existing graph tools typically require formal query languages (e.g., Cypher, SPARQL) or focus on manual exploration, limiting direct comparison. We also considered an ablation study, but the system relies on coordinated adaptive views that collectively support comparison tasks. Isolating components would not reflect realistic usage, as users rely on the interplay between natural language input, graph views, and summaries. Therefore, our evaluation focuses on how users interact with the system through measures of accuracy, interaction behavior, and gaze patterns.

Thus, the main study consisted of 18 statements in a Jeopardy-style fact-checking task [12, 41], where participants formulated their own queries. The statements were evenly distributed across three task types (*value comparison*, *path comparison*, and *neighborhood exploration*) and three comparison structures (*1–1*, *1–N*, and *N–N*), resulting in two statements for each task–structure combination. This design enabled systematic evaluation of the different comparison scenarios supported by NEXO. For each statement, participants used the *chat panel*, *graph view*, and *analysis and statistics* panel to determine whether the statement was correct or incorrect. Participants could submit multiple queries per task and were allowed to skip a statement after three attempts if they believed the system could not provide a sufficient answer. Task order was counterbalanced across participants to reduce ordering effects and learning bias.

We used the KG20C scholarly KG [47], which contains papers from 20 top-tier conferences represented as a multi-relational graph with five entity types and five relation types. The dataset was selected because it supports all three comparison task types: paper attributes and citations support value comparison, authorship and citation links support path comparison, and relationships among papers, venues, and domains support neighborhood exploration. This enables evaluation of all task categories using real-world scholarly data.

The study protocol was reviewed and approved by our institution's Institutional Review Board (IRB Protocol No. ET00048494), and all participants provided informed consent. Participants first received an overview of the system, its three interface panels, and the task structure, followed by a short guided tutorial on formulating natural language queries and interpreting system responses. We recruited 13 participants (ages 21–42, median = 23), including 6 self-reported males and 7 females. All participants had backgrounds in computer science or computer engineering.

Throughout the study, we collected task responses, completion times, natural language inputs, gaze data, and interaction logs. Gaze data were captured using a Pupil Labs Invisible eye tracker to measure visual attention across the interface. After completing the tasks, participants completed the SUS questionnaire [6] and post-study measures adapted from Hoffman et al. [19], including the *Explanation Satisfaction Scale* and *Trust Scale for the XAI context*. The session concluded with a short informal interview about participants' reasoning strategies and interaction patterns. Each session lasted approximately 60 minutes.

5.2 Interaction Patterns

As shown in Table 2, participants achieved high accuracy overall, with strong performance on value comparison (T1: 91.2%) and neighborhood exploration (T3: 92.3%), while path comparison (T2) resulted in lower accuracy (78.2%). This difference reflects the increased complexity of path-based tasks, which require participants to trace multi-step relationships, identify intermediate nodes, and compare sequences of connections rather than single values or local structures. In contrast, value and neighborhood tasks allow users to rely on more direct summaries or localized patterns, making them easier to interpret. Also, these results support *DG1*, which emphasizes the efficiency of natural language interaction for this exploration.

Despite these differences in task difficulty, participants required relatively few interactions to reach a decision. On average, participants issued between 1.10 and 1.36 queries per task across all task types, indicating that most queries were correctly interpreted and that users were able to obtain relevant results without extensive reformulation. This suggests that the combination of natural language input and coordinated visual feedback supports efficient exploration, even for more complex comparison tasks.

Additionally, Figure 8 and Figure 9 show how participants distributed their visual attention across the three interface panels during task completion which supports our *DG3*. Overall, participants spent

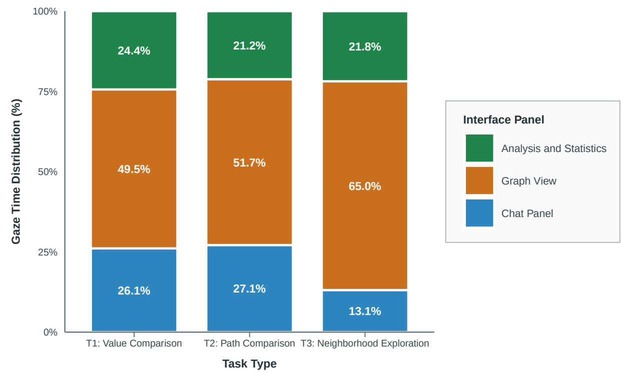

Figure 8: Gaze Time Distribution by Task Type. Mean proportion of viewing time for each component across three task types. Graph View dominated all tasks (49.5-65.0%), with T3 showing highest graph reliance.

Table 2: Summary of performance metrics across all participants (N=13).

Metric	**Mean**
System Usability	
Overall SUS Score	73.5
Average Accuracy across all 18 Trials	
T1: Value Comparison	91.2%
T2: Path Comparison	78.2%
T3: Neighborhood Exploration	92.3%
Average Number of Queries Submitted	
T1: Value Comparison	1.32
T2: Path Comparison	1.36
T3: Neighborhood Exploration	1.10

the majority of their gaze time on the graph view across all task types, reflecting its central role in reasoning about relationships in the KG. Specifically, the graph view accounted for approximately 49.5% of gaze time for T1 (value comparison), 51.7% for T2 (path comparison), and 65.0% for T3 (neighborhood exploration). In contrast, the chat panel and visualization panel received smaller but complementary attention, supporting query formulation and quantitative comparison respectively. Temporal analysis of gaze behavior further reveals how attention shifts during task progress. Early in the task, participants focused more on the chat panel to interpret the question and issue queries, while attention increasingly shifted toward the graph and visualization panels as participants explored relationships and validated results. This trend is particularly pronounced in T3 tasks, where the graph view dominates attention throughout the task, indicating that neighborhood exploration relies heavily on structural reasoning within the graph.

These patterns demonstrate that participants actively move between high-level summaries and detailed structural views, using different panels at different stages of the task. This behavior directly supports *DG3* by showing that the system enables users to maintain context while drilling down into detailed graph structures and, conversely, to return to summarized views for comparison and validation. The coordinated use of panels allows users to integrate structural, statistical, and textual information, supporting multi-level exploration (*DG2*) without overwhelming them with a single representation.

5.2.1 Post-Study Questionnaire Results

We also analyzed participants' post-study responses (N=13), assessing perceived usability, explanation quality, and natural language interaction. Overall, responses indicate positive perceptions of the system. The mean System Usability Scale (SUS) score was 73.5 (SD = 12.85). SUS scores have a range of 0 to 100, and a score of around 60 and above is generally considered as an indicator of good usability [24].

Table 3: Post-study questions (adapted from Hoffman et al. [19]) with ≥80% agreement (Agree/Strongly Agree).

Item (shortened)	*Agreement*
System is fast/efficient.	100%
I feel confident using the system.	92%
I understand how the system works.	92%
Explanations have sufficient detail.	92%
Explanations tell me how to use the system.	92%
Explanations are useful for my goals.	92%
NL queries were interpreted correctly.	85%
System outputs are predictable.	85%
Explanations are satisfying.	85%
Explanations feel complete.	85%
Explanations reflect system accuracy.	85%

Additionally, responses in Table 3 indicate high agreement on system efficiency (100%) and user confidence (92%), suggesting that participants trusted the system to support their analysis. Participants also reported strong agreement regarding explanation quality and completeness (92%–85%), indicating that system outputs were perceived as understandable, reliable, and sufficient for decision-making.

5.2.2 Post-Task Interview

Following the task, we conducted semi-structured interviews with participants to gather qualitative data on their interaction strategies and experiences with the system. Participants were asked open-ended questions such as Which panel helped you make decisions?, What challenges or strengths did you experience during the interaction?, and their overall impressions of using natural language queries in combination with multiple coordinated views.

Overall, participants highlighted the complementary roles of the three panels in supporting comparison tasks. In particular, 5 out of 13 participants noted that for straightforward value comparisons, concise summaries in the chat or visualization panel were sufficient, reducing the need to inspect detailed graph structures. As one participant stated, *"For simple comparisons, the text summary was enough, I didn't need to look at all the details unless I wanted to double-check where the values came from and make sure it wasn't hallucinating."* Additionally, 4 participants emphasized the system's ability to retrieve and organize comparative results, especially in group-based queries. For example, one participant noted, *"I like how fast the system could expand and show all combinations, like authors affiliated with a university, and then compare them side by side without me having to manually search."* Finally, 2 participants mentioned interaction features such as hover-based details and highlighting as beneficial for understanding complex structures, particularly in path comparison tasks. One participant commented, *"I liked the hover interactions as it made it easy to see details of each node, especially when there were many nodes in a path."*

6 Discussion

Beyond demonstrating the feasibility of natural language interaction for KG exploration, our work contributes a comparison-oriented perspective on visual analytics for KGs. Rather than focusing solely on query generation, NEXO treats natural language as a mechanism for expressing comparison intent and couples this interpretation with coordinated adaptive visual representations designed around comparison tasks (Figure 1). The study results suggest that users rely on both the transparent parameter interpretation process and the task-adaptive visualizations to understand, verify, and compare graph data. This design is motivated by prior findings that working with KGs involves challenges in querying, interpreting large graph structures, and extracting relevant insights efficiently [18, 22]. In particular, prior work reports difficulties in querying KGs, limited support for organic discovery, and overly complex visualizations for end users [11, 14, 22]. Our approach addresses this by allowing users to express comparison tasks in natural

Figure 9: Temporal Attention Allocation Patterns. Evolution of gaze distribution across task periods showing systematic shifts from graph-based exploration (First Period) to visualization-based analysis (Third Period). Shaded regions represent 95% confidence intervals.

language while exposing structured parameters that can be inspected and refined when needed, consistent with prior work on natural language interfaces for data exploration [28, 35, 39]. In addition, prior research has shown that node-link visualizations often struggle with scalability and readability in large graphs, limiting their effectiveness for analysis and comparison [18, 22]. To mitigate this, our system combines graph views with summary and statistical representations, aligning with established principles of coordinated multiple views for balancing detail and overview [34, 48].

As shown in Table 2, participants achieved high accuracy across tasks (T1: 91.2%, T2: 78.2%, T3: 92.3%), indicating that the system supports comparison tasks reliably across different structures. At the same time, participants required only 1.10–1.36 queries per task (Table 2), suggesting that the natural language interface reduced the need for repeated query reformulation. However, this efficiency does not imply that a single representation was sufficient. Instead, both gaze data (Figure 8) and qualitative responses indicate that participants distributed their reasoning across panels. Specifically, the graph view received the largest proportion of attention across all tasks (49.5% for T1, 51.7% for T2, and 65.0% for T3), showing that users relied on it for structural reasoning. In contrast, post-study responses (Table 3) and interview feedback indicate that participants used the analysis panel and textual summaries to confirm results and reduce effort when possible. This suggests that supporting comparison in KGs requires coordinated representations, where each view contributes differently rather than replacing one another.

The results also reveal limitations in how effectively these representations are coordinated during interaction. Trends in Figure 9 show that panel usage changes over time depending on task demands. For neighborhood exploration (T3), participants initially relied heavily on the graph view (77.6% gaze time), indicating that understanding local structure requires direct inspection of nodes and edges. However, this reliance decreases over time while attention to the analysis and statistics panel increases (Figure 9), suggesting that the graph becomes harder to manage as more information is revealed. This shift indicates that the graph view alone does not scale well for sustained comparison, and users transition to summarized representations to reduce cognitive load. In contrast, for value comparison (T1), participants focused more on the chat panel early in the task (Figure 9), which aligns with qualitative feedback where multiple participants reported that concise textual summaries were sufficient for simple comparisons. This behavior suggests that presenting full graph structure in such cases may introduce unnecessary complexity. Together, these findings indicate that while the system provides multiple levels of representation, it does not actively adapt which view should be emphasized based on task complexity or interaction stage. As a result, users must decide when to switch between detailed and summarized views themselves, which introduces coordination overhead and may limit efficiency in more complex analytical scenarios.

A pattern observed in our study is that participants received limited guidance on how to navigate between representations during comparison tasks. While the system provides multiple coordinated views, it does not indicate which panel may be most effective at different stages of analysis. This is reflected in the gaze transitions shown in Figure 9, where users independently shifted from the graph view to the analysis panel as task complexity increased. This suggests that participants had to infer effective interaction strategies on their own, although such overhead may decrease with experience. Additionally, the lower accuracy observed in path comparison tasks (78.2% in Table 2) indicates that multi-step relational reasoning remains challenging despite visual and textual support. This suggests that presenting paths alone may be insufficient and that users could benefit from additional abstractions or guidance for comparing intermediate entities and relationships. Finally, qualitative feedback revealed that participants frequently used the graph view to verify results, highlighting both the importance of transparency and limited trust in summarized outputs alone.

Our evaluation focuses on a single KG dataset (KG20C [47]), selected to provide a controlled yet structurally rich environment for assessing the full system pipeline. KG20C includes multiple entity and relation types (e.g., authors, papers, citations, venues, and domains), enabling all three comparison task types (T1–T3) within a unified setting. This allows us to evaluate how natural language queries, parameter extraction, and coordinated visualizations interact without confounding differences in schema, data quality, or domain-specific semantics. Nevertheless, the proposed framework is organized around domain-independent comparison operations. Value comparison focuses on graph entity attributes, path comparison on relationships and connectivity between entities, and neighborhood exploration on local graph structure. Similarly, the 1–1, 1–N, and N–N comparison structures describe how graph elements are compared rather than what they represent. As a result, the framework can be applied to other knowledge graph domains, such as biomedical knowledge graphs (e.g., genes and diseases) and social networks (e.g., users, communities, and interactions), among others, where users compare attributes, relationships, and local graph structures despite differences in schema and semantics. Furthermore, because NEXO focuses on task-relevant entities, paths, and neighborhoods rather than full-graph visualization, the interaction framework may also support exploration in larger graphs by restricting analysis to graph regions most relevant to the comparison task.

7 Future Work and Conclusion

We presented NEXO, a comparison-driven KG exploration system that integrates natural language interaction with coordinated adaptive visualizations. To address challenges in querying and interpreting complex graph data, NEXO maps natural language input to structured comparison tasks and adaptively presents results across multiple views. Through example scenarios, we illustrated how users can explore and compare KG data using textual summaries, structural graph views, and statistical representations. We also reported results from a user study, showing how users distribute attention across views based on task demands. These findings suggest opportunities for improving cross-view guidance and supporting more complex comparison tasks in future KG visualization systems.

Future work can explore adaptive interaction mechanisms that guide users toward effective representations based on task type and interaction stage. For example, the system could dynamically emphasize or recommend specific panels as graph complexity increases, reducing coordination overhead in multi-view analytical workflows. Another direction is supporting more complex relational reasoning, particularly for multi-step paths and larger neighborhoods. The lower accuracy observed in path comparison tasks suggests these scenarios remain challenging and require additional support. While NEXO exposes underlying graph structures, users must still interpret intermediate relationships and compare them manually. Future work could introduce higher-level abstractions, such as path summaries, intermediate-node grouping, or automated highlighting of key subgraph differences, to reduce cognitive effort and make comparisons more explicit.

REFERENCES

[1] J. Abello, F. van Ham, and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006. doi: 10.1109/TVCG.2006.120 3

[2] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 483–492. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2470654.2470724 2

[3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117. IEEE, 2005. doi: 10.1109/INFVIS.2005.1532136 3, 4

[4] V. Batagelj, F. J. Brandenburg, W. Didimo, G. Liotta, P. Palladino, and M. Patrignani. Visual analysis of large graphs using (x,y)-clustering and hybrid visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1587–1598, 2011. doi: 10.1109/TVCG.2010.265 3

[5] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185 5, 7

[6] J. Brooke. Sus: a retrospective. *Journal of usability studies*, 8(2), 2013. 7

[7] D. V. Camarda, S. Mazzini, and A. Antonuccio. Lodlive, exploring the web of data. In *Proceedings of the 8th International Conference on Semantic Systems*, pp. 197–200, 2012. doi: 10.1145/2362499.2362532 1

[8] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, p. 69. IEEE Computer Society, USA, 2000. 4

[9] K. Dhamdhere, K. S. McCurley, R. Nahmias, M. Sundararajan, and Q. Yan. Analyza: Exploring data with conversation. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pp. 493–504, 2017. doi: 10.1145/3025171.3025227 4

[10] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson. Graphtrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1663–1672, 2012. doi: 10.1145/2207676.2208293 2

[11] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault et al. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pp. 1433–1445. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3183713.3190657 8

[12] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th annual acm symposium on user interface software & technology*, pp. 489–500, 2015. doi: 10.1145/2807442.2807478 1, 2, 4, 6, 7

[13] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011. doi: 10.1177/1473871611416549 1, 2, 3

[14] S. Harris and A. Seaborne. Sparql 1.1 query language. W3C Recommendation, Mar. 2013. 8

[15] P. Heim, S. Hellmann, J. Lehmann, S. Lohmann, and T. Stegemann. Relfinder: Revealing relationships in rdf knowledge bases. In *International Conference on Semantic and Digital Media Technologies*, pp. 182–187. Springer, 2009. doi: 10.1007/978-3-642-10543-2_21 1

[16] N. Heist, S. Hertling, D. Ringler, and H. Paulheim. Knowledge graphs on the web-an overview. *Knowledge Graphs for eXplainable Artificial Intelligence*, pp. 3–22, 2020. 1

[17] N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302–1309, 2007. doi: 10.1109/TVCG.2007.70582 2

[18] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on visualization and computer graphics*, 6(1):24–43, 2002. doi: 10.1109/2945.841119 4, 8, 9

[19] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman. Measures for explainable ai: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-ai performance. *Frontiers in Computer Science*, 5:1096257, 2023. doi: 10.3389/fcomp.2023.1096257 7, 8

[20] C.-W. Hsuan Yuan, T.-W. Yu, J.-Y. Pan, and W.-C. Lin. Kgscope: Interactive visual exploration of knowledge graphs with embedding-based guidance. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7702–7716, 2024. doi: 10.1109/TVCG.2024.3360690 1

[21] B. Lee, C. S. Parr, C. Plaisant, B. B. Bederson, V. D. Veksler, W. D. Gray et al. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426, 2006. doi: 10.1109/TVCG.2006.106 1, 2, 3

[22] H. Li, G. Appleby, C. D. Brumar, R. Chang, and A. Suh. Knowledge graphs in practice: Characterizing their users, challenges, and visualization opportunities. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):584–594, 2023. doi: 10.1109/TVCG.2023.3326904 1, 2, 8, 9

[23] H. Li, Y. Wang, S. Zhang, Y. Song, and H. Qu. Kg4vis: A knowledge graph-based approach for visualization recommendation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):195–205, 2021. doi: 10.1109/TVCG.2021.3114863 1

[24] V. Lopez, M. Fernández, E. Motta, and N. Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic web*, 3(3):249–265, 2012. doi: 10.3233/SW-2011-0030 8

[25] Y. Munarko, A. Rampadarath, and D. P. Nickerson. Casbert: Bert-based retrieval for compositely annotated biosimulation model entities. *Frontiers in Bioinformatics*, 3:1107467, 2023. doi: 10.3389/fbinf.2023.1107467 1

[26] T. Munzner. *Visualization analysis and design*. CRC press, 2014. 2

[27] R. Nararatwong, N. Kertkeidkachorn, and R. Ichise. Knowledge graph visualization: Challenges, framework, and implementation. In *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 174–178. IEEE, 2020. doi: 10.1109/AIKE48582.2020.00034 1

[28] A. Narechania, A. Srinivasan, and J. Stasko. Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):369–379, 2020. doi: 10.1109/TVCG.2020.3030378 1, 2, 4, 9

[29] OpenAI. Chatgpt, 2025. Large language model. 6

[30] F. Orlandi, J. Debattista, I. A. Hassan, C. Conran, M. Latifi, M. Nicholson et al. Leveraging knowledge graphs of movies and their content for web-scale analysis. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 609–616. IEEE, 2018. doi: 10.1109/SITIS.2018.00098 1

[31] S. Oviatt, R. Coulston, S. Tomko, B. Xiao, R. Lunsford, M. Wesson et al. Toward a theory of organized multimodal integration patterns during human-computer interaction. In *Proceedings of the 5th international conference on Multimodal interfaces*, pp. 44–51, 2003. doi: 10.1145/958432.958443 1

[32] M. G. Ozsoy, L. Messallem, J. Besga, and G. Minneci. Text2cypher: Bridging natural language and graph databases. In *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, pp. 100–108, 2025. 2

[33] C. Partl, S. Gratzl, M. Streit, A. M. Wassermann, H. Pfister, D. Schmalstieg et al. Pathfinder: Visual analysis of paths in graphs. In *Computer Graphics Forum*, vol. 35, pp. 71–80. Wiley Online Library, 2016. doi: 10.1111/cgf.12883 2

[34] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory

visualization. In *Fifth international conference on coordinated and multiple views in exploratory visualization (CMV 2007)*, pp. 61–71. IEEE, 2007. doi: 10.1109/CMV.2007.20 4, 9

[35] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th annual symposium on user interface software and technology*, pp. 365–377, 2016. doi: 10.1145/2984511.2984588 1, 2, 9

[36] R. Shahriari, A. Hashky, S. Arya, T. Audino, E. D. Ragan, V. Gogate et al. Comparison of text-based inputs for human-in-the-loop feedback in vision–language models. *ACM Trans. Interact. Intell. Syst.*, May 2026. doi: 10.1145/3816700 6

[37] R. Shahriari, E. D. Ragan, and J. Ruiz. Natural language interaction for editing visual knowledge graphs. In *Proceedings of the 13th Knowledge Capture Conference 2025*, pp. 26–34, 2025. doi: 10.1145/3731443.3771344 1

[38] R. Shahriari, Y. Yang, D. N. A. Tamboli, M. Perez, Y. Zha, J. Hou et al. Muchex: A multimodal conversational debugging tool for interactive visual exploration of hierarchical object classification. *IEEE Computer Graphics and Applications*, 2025. doi: 10.1109/MCG.2025.3598204 4

[39] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang et al. Towards natural language interfaces for data visualization: A survey. *IEEE transactions on visualization and computer graphics*, 29(6):3121–3144, 2022. doi: 10.1109/TVCG.2022.3148007 1, 2, 4, 9

[40] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pp. 364–371. Elsevier, 2003. doi: 10.1016/B978-155860915-0/50046-9 3, 4

[41] A. Srinivasan and J. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE transactions on visualization and computer graphics*, 24(1):511–521, 2017. doi: 10.1109/TVCG.2017.2745219 1, 2, 6, 7

[42] B. Steichen, G. Carenini, and C. Conati. User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pp. 317–328, 2013. doi: 10.1145/2449396.2449439 4

[43] M. Thattai and A. Van Oudenaarden. Intrinsic noise in gene regulatory networks. *Proceedings of the National Academy of Sciences*, 98(15):8614–8619, 2001. doi: 10.1073/pnas.151588598 1

[44] A. Theocharidis, S. Van Dongen, A. J. Enright, and T. C. Freeman. Network visualization and analysis of gene expression data using biolayout express 3d. *Nature protocols*, 4(10):1535–1550, 2009. doi: 10.1038/nprot.2009.177 1

[45] C. Tominski, J. Abello, F. van Ham, and H. Schumann. Fisheye tree views and lenses for graph visualization. In *Tenth International Conference on Information Visualisation (IV'06)*, pp. 17–24, 2006. doi: 10.1109/IV.2006.54 3

[46] C. Tominski, C. Forsell, and J. Johansson. Interaction support for visual comparison inspired by natural behavior. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2719–2728, 2012. doi: 10.1109/TVCG.2012.237 2

[47] H.-N. Tran and A. Takasu. Kg20c & kg20c-qa: Scholarly knowledge graph benchmarks for link prediction and question answering. *arXiv preprint arXiv:2512.21799*, 2025. 5, 7, 9

[48] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 110–119, 2000. doi: 10.1145/345513.345271 4, 9

[49] F. Yanez, C. Conati, A. Ottley, and C. Nobre. The state of the art in user-adaptive visualizations. In *Computer Graphics Forum*, vol. 44, p. e15271. Wiley Online Library, 2025. doi: 10.1111/cgf.15271 4

[50] B. Yu and C. T. Silva. Flowsense: A natural language interface for visual data exploration within a dataflow system. *IEEE transactions on visualization and computer graphics*, 26(1):1–11, 2019. doi: 10.1109/TVCG.2019.2934668 2

[51] M. Zhu, W. Chen, Y. Hu, Y. Hou, L. Liu, and K. Zhang. Drgraph: An efficient graph layout algorithm for large-scale graphs by dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1666–1676, 2020. doi: 10.1109/TVCG.2020.3030447 1